PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

51) International Patent Classification 6 :
H04L 9/00

A1

(11) International Publication Number: WO 98/36520

(43) International Publication Date: 20 August 1998 (20.08.98)

21) International Application Number: PCT/US97/11304

22) International Filing Date: 20 June 1997 (20.06.97)

30) Priority Data:
60/039,696 13 February 1997 (13.02.97) US

71) Applicant *(for all designated States except US)*: SECURE TRANSACTION SOLUTIONS, LLC [US/US]; Suite 220, 1953 Gallows Road, Vienna, VA 22182 (US).

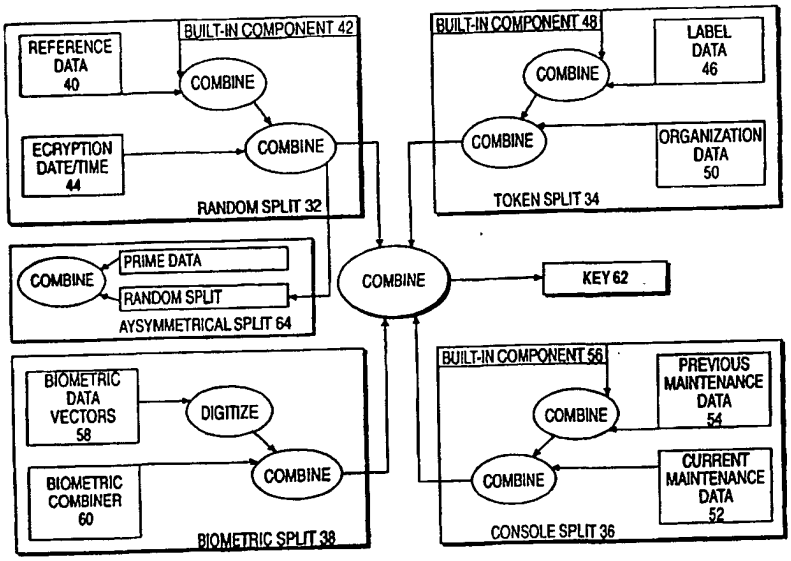
72) Inventors; and
75) Inventors/Applicants *(for US only)*: SCHEIDT, Edward, M. [US/US]; 1048 Deep Run Drive, McLean, VA 22101 (US). WACK, C., Jay [US/US]; 13715 Lewisdale Road, Clarksburg, MD 20871 (US).

74) Agent: CHAMPAGNE, Thomas, M.; Rabin, Champagne & Lynt, P.C., Suite 1111, 1725 K Street, N.W., Washington, DC 20006 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, US, UZ, VN, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: CRYPTOGRAPHIC KEY SPLIT COMBINER



(57) Abstract

A random key split (32) may be made randomly or pseudorandomly. A token-stored key split may be generated (34). In addition, a console-stored key split may be generated (36) and a biometric-source-provided key split may be used. In the random key split (32), the split is randomly or pseudorandomly generated based upon a seed which is provided by reference data (40). Alternatively, the seed may be combined or randomized with a built-in component (42). Further randomization may be obtained by using the date and time of encryption (44).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

CRYPTOGRAPHIC KEY SPLIT COMBINER

Technical Field

The present invention relates to cryptographic systems. In particular, the present invention relates to a system for formulating cryptographic keys used to encrypt plaintext messages or embedded objects and decrypt ciphertext communications media.

Background Art

In the modern world, communications are passed between parties in a variety of different ways utilizing many different communications media. Electronic communication is becoming increasingly popular as an efficient manner of transferring information, and electronic mail in particular is proliferating due to the immediacy of the medium. Another communications medium at the software program level defines an object as a particular piece of compiled code that provides a specific service within the overall system.

Unfortunately, drawbacks accompany the benefits provided by electronic communication, particularly in the area of privacy. Electronic communications may be intercepted by unintended recipients. Wireless transmissions, such as voice communication by cellular telephone, and electronic mail are especially susceptible to such interception. Also, the retention of information on a computing system may raise other privacy issues. Multiple users on a common computing device and separation of information for multiple applications for a network of users communicating different categories of information are among the scenarios for which privacy may be a concern. In another context, the idea of privacy may extend beyond keeping information from prying eyes; the integrity of software program objects may be a concern. The manipulation or other modification of an object can cause results unintended by the creator of the object.

The problem of electronic communication privacy has been addressed, and solutions to the problem have been put in place. One form of solution uses cryptography to provide privacy for electronic communication. Cryptography involves the encrypting or encoding of a transmitted or stored message or object, followed by the decryption or decoding of a received or retrieved message or object. The message or object usually takes the form of a digital signal, a digitized analog signal, or a functionality of the object. If the communication is intercepted during transmission or is extracted from storage by an unauthorized entity, the message is worthless to the interloper, who does not possess the means to decrypt the encrypted message.

In a system utilizing cryptography, the encrypting side of the communication incorporates an encoding device or encrypting engine. The encoding device accepts the plaintext (unencrypted) message (or object) and a cryptographic key, and encrypts the plaintext message (or object) with the key according to an encrypt relation that is predetermined for the plaintext communication and the key. That is, the message or object is manipulated with the key in a predetermined manner set forth by the text/key relation to produce a ciphertext (encrypted) message or object.

Likewise, the decrypting side of the communication incorporates a decoding device or decrypting engine. The decoding device accepts the ciphertext message (or object) and a cryptographic key, and decrypts the ciphertext message with the key according to a decrypt relation that is predetermined for the ciphertext message (or object) and the key. That is, the message (or object) is manipulated with the key in a predetermined manner set forth by the text/key relation to produce a new plaintext message that corresponds with the original plaintext message.

The manner in which the key and the relation are applied in the communication process, and the manner in which keys are managed, define a cryptographic scheme. There are many conventional cryptographic schemes in use today. For example, probably the most popular of these is a public-key cryptographic scheme. According to a scheme of this type, the keys used are actually combinations of a public key component that is available to anyone or

to a large group of entities, and a private key component that is specific to the particular communication.

An important consideration in determining whether a particular cryptographic scheme is adequate for the application is the degree of difficulty necessary to defeat the cryptography, that is, the amount of effort required for an unauthorized person to decrypt the encrypted message. One way to improve the security of the cryptographic scheme is to minimize the likelihood that a valid key can be stolen, calculated, or discovered. The more difficult it is for an unauthorized person to obtain a valid key, the more secure communications will be under a particular scheme.

Disclosure of the Invention

It is therefore an objective of the present invention to provide a process and apparatus for assembling keys which provides added security against compromising a communications medium, which may include software component objects, by unauthorized entities.

It is a further objective of the present invention to provide a process and apparatus for developing key components that cannot be reproduced by unauthorized parties.

These and other objectives and advantages are provided by a cryptographic key split combiner, which includes a number of key split generators for generating cryptographic key splits and a key split randomizer for randomizing the cryptographic key splits to produce a cryptographic key. Each of the key split generators generates key splits from seed data. The source of the seed data can be a pseudorandom or random data sequence that may be included in a key management scheme that uses the key splits for determining the data cryptographic or session key. The management of the key splits can include provision of a source for the seed data and a distribution process to ensure that the desired combination of key splits is generated.

In one embodiment of the present invention, the key split generators include a random split generator for generating a random key split based on reference data. The random split generator may generate a random sequence

based on the reference data, or may generate a pseudorandom sequence based on the reference data. The random key split may further be based on chronological data. The random key split may instead be based on the reference data and on static data, which may be updated. One manner of updating the static data is by modifying a prime number divisor of the static data.

Other key split generators may include, for example, a token split generator for generating a token key split based on label data and/or organization data and/or static data; a console split generator for generating a console key split based on maintenance data, whether previous or current, and/or on static data; an asymmetrical key split generator for generating pair-wise data; and a biometric split generator for generating a biometric key split based on biometric data, which may include biometric data vectors and on biometric combiner data, and/or static data. The label data may be associated with label categories and sub-categories of addressees, which are meaningful to a user who is specifying or determining the intended recipients(s) of the encrypted information or object. The label data may be read from a storage medium, and may include user authorization data. The resulting cryptographic key may be, for example, a stream of symbols, at least one symbol block, or a key matrix.

An asymmetrical key split generator may be used to ensure the integrity of one or more of the key split generators, such as the random key split, or to ensure the integrity of the sender's data.

The key split generators may be used to determine which, if any, methods and properties are allowed in a software program that includes component objects. A component object is a compiled piece of software code in computer memory, which has an array of memory addresses, and indicates relatively where in memory certain functions or methods and data or properties of that object are stored. An array associated with the component object may use key splits which determine which methods and properties are allowed and control access to the memory address for those allowed methods and properties.

The present invention also includes a process for forming cryptographic or session keys, which includes generating a plurality of cryptographic key splits from seed data and randomizing the cryptographic key splits to produce a

cryptographic key. The process can include generating reference pointers to the key splits that would facilitate the selection of key splits during the encrypting or decrypting process. Once the data or object is encrypted, these pointers can be included with the ciphertext.

The cryptographic key splits may include, for example, a random key split based on reference data, a token key split based on label data, a console key split based on maintenance data, and a biometric key split based on biometric data. These key splits may be random sequences or pseudorandom sequences.

Generating the random key split may include generating a key split based on the reference data and on chronological data, or based on the reference data and on static data. Generating the token key split may include generating a key split based on the label data, which may be read from a storage medium and may include authorization data, and on organization data, or based on the label data and on static data. Generating the console key split may include generating a key split based on previous maintenance data and on current maintenance data, or based on the maintenance data and on static data. Generating the biometric key split may include generating a key split based on biometric data vectors and on biometric combiner data, or based on the biometric data and on static data.

The static data provided for any of the key splits may be updated. Updating the static data may include modifying a prime number divisor of the static data.

The resulting cryptographic or session key may be a stream of symbols, at least one symbol block, or a key matrix.

Brief Description of Drawings

The present invention will be more completely understood by way of the following detailed description, with reference to the following drawings, wherein:

FIG. 1 shows a block diagram of a communications event featuring cryptography.

FIG. 2 is a block diagram of a key split combiner.

FIG. 3 is an exemplary hardware implementation of the key generation aspect of the present invention.

Best Mode for Carrying Out the Invention

Referring to FIG. 1, a communication has an origination space 2 and a destination space 4. The origination space 2 defines the place and time at which the communication originates. The destination space 4 defines the place and time at which the communication is intended to be decoded. The origination space 2 and the destination space 4 may be remote in location. Alternatively, they may be collocated but displaced in time. The space and time correspondence between the origination space 2 and the destination space 4 depends on the nature of a particular communication. The origination space 2 and destination space 4 are coupled to a common communications channel 6. This communications channel 6 may bridge a physical space, such as empty air in the case of a cellular voice telephone call. Alternatively, the communications channel 6 may be temporary storage for the communication while time passes between the origination space 2 and the destination space 4, such as a message left in memory on a computer by a first user, for a second user to read at a later time on the same computer. The communications channel 6 may also be a combination of the two, such as telephone cables and storage memory in the case of an electronic mail transmission. The communications channel 6 may also be a component object in computer memory.

A component object is a compiled piece of software code in computer memory, which has an array of memory addresses, and indicates relatively where in memory certain functions or methods and data or properties of that object are stored. An application programmer makes use of the component object by obtaining a pointer to the memory that contains the array. This is known in the art as creating an instance of a component object. The programmer can then make use of the methods and properties of the component object by indirectly addressing them via the array.

At the origination space 2, the original plaintext message 8 is received and encrypted according to the encrypt text/key relation 14, using a provided encrypt

key 10, to create a ciphertext message 16. The ciphertext message 16 is received at the destination space 4 via the communications channel 6. An authorized entity having a proper decrypt key 20 can then provide the decrypt key 20 to the destination space 4, where it is applied to the ciphertext message 16 according to a decrypt text/key relation 22 to create a new plaintext message 24 which corresponds to the original plaintext message 8.

The origination space 2 and the destination space 4 can be, for example, computers, or even the same computer. An exemplary computer may have a certain amount of storage space in the form of memory for storing the text/key relation. A microprocessor or similar controller, along with a control structure and random access memory for storing original plaintext and keys provided by a user, can be included in each space and can perform the functions of the encryption/decryption engine. An input device 26, 28, such as a keyboard, floppy disk drive, CD-ROM drive, or biometrics reader, can also be provided for accepting the key and plaintext message from the origination user, and the key from the destination user. At the destination space 4, an output device 30, such as a monitor, disk drive, or audio speaker, may also be provided to present the new plaintext message to the destination user. The text/key relation can be stored on a floppy disk or other permanent or temporary portable storage, rather than in hard storage in the computer, to allow different text/key relations to be applied by different users or in different situations.

The keys that are provided at the origination space and at the destination space may be composed of several components, or splits, each of which may be provided by a different source. As shown in Fig. 2, a random key split 32 may be randomly or pseudorandomly generated. A second split 34 may be stored on a token. A third split 36 may be stored on a console, and a fourth split 38 may be provided by a biometric source. The key splits may be combined to form a complete cryptographic key. This key may take the form of a stream of symbols, a group of symbol blocks, an N-dimensional key matrix, or any other form usable by the particular encryption scheme.

The random split 32 provides a random component to the cryptographic key. This split 32 is randomly or pseudorandomly generated based on a seed

which is provided by any source as reference data 40. For example, when a user attempts to log on to a system, the date and time of the user's log-on attempt, represented in digital form, can be used as a seed to generate the key split. That is, the seed may be provided to a pseudorandom sequence generator or other randomizer to produce the random split. Such pseudorandom sequence generators are well known in the art. For example, a simple hardware implementation could include a shift register, with various outputs of the register XORed and the result fed back to the input of the register. Alternatively, the seed may be combined, or randomized, with a built-in component 42, such as a fixed key seed stored at the origination space. The randomization may be performed, for example, by applying a variation of the text/key relation to the generated seed and the stored fixed key seed. This result may be further randomized with, for example, a digital representation of the date and time of the encryption 44, in order to produce the random key split 32.

The token split 34 may be generated in a similar fashion. In this case, the seed is provided on a token, that is, it is stored on a medium that is possessed by the user. For example, the seed may be stored on a floppy disk that the system must read as part of the encryption procedure. The token may store a number of different seeds, or label data 46, each of which corresponds to a different authorization provided by the system or specified by the user. For example, one seed may be used to generate a key split to authorize a particular user to read a message at a particular destination space. Another key seed may be used to generate a key split to authorize any member of a group of users to read a message at any destination space, and for one particular user to read the message and write over the message at a particular destination space. The label data 46 may even designate a window of time during which access to the communication is valid. This seed may be randomized with a built-in component 48, such as a seed stored at the origination space, which may then be further randomized with organization data 50 provided to the organization to which the user belongs.

The console split 36 is derived from a changing value stored at a user space, such as on a system console. Maintenance data, such as the checksum taken from a defragmentation table set, may be used to produce such changing values. For example, the current maintenance data 52 may be randomized with particular previous maintenance data. Alternatively, all previous maintenance data 54 may be randomized with a built-in component 56 stored at the origination space, the results of which are XORed together and randomized with the current maintenance data 52. The randomization result of the changing value is the console split 36.

The biometric split 38 is generated from biometric data vectors 58 provided by biometric samples of the user. For example, a retinal scanner may be used to obtain a unique retinal signature from the user. This information, in digital form, will then be used to generate the biometric split 38. This may be accomplished by, for example, randomizing a digital string corresponding to the biometric vectors 58 with biometric combiner data 60, which may be a digital hash of the user's system identification number or some other identifying data that can be linked to the user's physical data provided by the biometric reader. The resulting randomized data is the biometric split 38. The biometric split 38 provides information that is incapable of being reproduced by anyone but the user providing the biometric data vector 58.

The built-in key split components 42, 48, 56 described herein may be static in that they do not change based on uncontrolled parameters within the system. They may be updated for control purposes, however. For example, the built-in key split components 42, 48, 56 may be changed to modify the participation status of a particular user. The key split component may be changed completely to deny access to the user. Alternatively, only a single prime number divisor of the original key split component may be taken from the key split component as a modification, in order to preserve a legacy file. That is, the user will be able to access versions of the file created prior to the modification, but will not be allowed to change the file, effectively giving the user read-only access. Likewise, modification of the key split component can be effected to grant the user broader access.

According to one cryptographic scheme that may be used in accordance with the present invention, a prime number and a random number are generated from a data seed source for one or more of the communicating parties. The random number can be used in the "public" domain, such as on a public server, or may be negotiated between the parties prior to the communications process. To establish communications between two parties, a polynomial or modulo calculation is made of the sender's prime number and the recipient's random number for the sender. The recipient calculates the recipient's prime number and the sender's random number. The two-way calculation creates a cryptographic or session key that is used to encrypt the random key split or encrypt a hash of the transmitted or stored message, thereby creating an asymmetrical split 64. The other key split generators that are used for the encrypting side of the communications provide integrity to the asymmetrical key split generator.

Once the key splits 32, 34, 36, 38 have been generated, they may be randomized together to produce the cryptographic key 62 for the communication. In performing each combination to generate the complete cryptographic key, a different variation of the text/key relation may be applied. The use of a plurality of different text/key relation variations adds to the security of the overall cryptographic scheme. It is contemplated that key splits other than those specifically described herein may be combined in forming the complete key 62. The total number of splits may also vary, and these splits may be used to build a key matrix to add to the complexity of the system. This complete key 62 should be in a form suitable for use in the particular cryptographic scheme. That is, different fields in the key may have different functions in the protocol of the communication, and should be arranged accordingly within the key.

At the destination space, the process is reversed in order to determine whether a user attempting to access a message has authorization, that is, has the valid key. The key supplied by the user at the destination space must include information required by the labels that were used to create the token split at the origination space. This information may also take the form of a token split. Further, a biometric split may be required as part of the destination key, in

order to provide a link between assigned identification data for the user and physical data collected from the user biometrically. The token split and the biometric split may be combined with other splits at the destination space to form the complete destination key.

FIG. 3 shows an exemplary hardware implementation for generating and managing the keys according to the present invention.

In the case of component object control, the array of addresses can be encrypted in the executable file of the component object. The application program using the component object can then call a special "create instant" function to pass along key splits or label representations. The "create instant" will: 1) using the key splits, determine which, if any, methods and properties are allowed , based on the passed key splits; 2) decrypt the memory address for those allowed methods and properties; and 3) modify the addresses of the methods and properties that are not allowed, thereby to instead call a "stub" function which will return an error code corresponding to the determination of no authorization. Note that there is no attempt to encrypt application data as it is passed to and from the component object.

The invention has been described using exemplary and preferred embodiments. However, the scope of the present invention is not limited to these particular disclosed embodiments. To the contrary, the present invention is contemplated to encompass various modifications and similar arrangements The scope of the claims, therefore, should be accorded the broadest interpretation so as to include all such modifications and similar arrangements.

What is claimed is:

1. A cryptographic key split combiner, comprising:

a) a plurality of key split generators for generating cryptographic key splits; and

b) a key split randomizer for randomizing the cryptographic key splits to produce a cryptographic key;

c) wherein each of said key split generators includes means for generating key splits from seed data.

2. The cryptographic key split combiner of claim 1, wherein said plurality of key split generators includes a random split generator for generating a random key split based on reference data.

3. The cryptographic key split combiner of claim 2, wherein said random split generator includes means for generating a random sequence based on the reference data.

4. The cryptographic key split combiner of claim 2, wherein said random split generator includes means for generating a pseudorandom sequence based on the reference data.

5. The cryptographic key split combiner of claim 2, wherein said random split generator includes means for generating a key split based on the reference data and on chronological data.

6. The cryptographic key split combiner of claim 2, wherein said random split generator includes means for generating a key split based on the reference data and on static data.

7. The cryptographic key split combiner of claim 6, further including means for updating the static data.

8. The cryptographic key split combiner of claim 7, wherein the means for updating the static data includes means for modifying a prime number divisor of the static data.

9. The cryptographic key split combiner of claim 1, wherein said plurality of key split generators includes a token split generator for generating a token key split based on label data.

10. The cryptographic key split combiner of claim 9, further comprising means for reading the label data from a storage medium.

11. The cryptographic key split combiner of claim 9, wherein the label data includes user authorization data.

12. The cryptographic key split combiner of claim 9, wherein said token split generator includes means for generating a random sequence based on the label data.

13. The cryptographic key split combiner of claim 9, wherein said token split generator includes means for generating a pseudorandom sequence based on the label data.

14. The cryptographic key split combiner of claim 9, wherein said token split generator includes means for generating a key split based on the label data and on organization data.

15. The cryptographic key split combiner of claim 9, wherein said token split generator includes means for generating a key split based on the label data and on static data.

16. The cryptographic key split combiner of claim 15, further including means for updating the static data.

17. The cryptographic key split combiner of claim 16, wherein the means for updating the static data includes means for modifying a prime number divisor of the static data.

18. The cryptographic key split combiner of claim 1, wherein said plurality of key split generators includes a console split generator for generating a console key split based on maintenance data.

19. The cryptographic key split combiner of claim 18, wherein said console split generator includes means for generating a random sequence based on the maintenance data.

20. The cryptographic key split combiner of claim 18, wherein said console split generator includes means for generating a pseudorandom sequence based on the maintenance data.

21. The cryptographic key split combiner of claim 18, wherein said console split generator includes means for generating a key split based on previous maintenance data and on current maintenance data.

22. The cryptographic key split combiner of claim 18, wherein said console split generator includes means for generating a key split based on the maintenance data and on static data.

23. The cryptographic key split combiner of claim 22, further including means for updating the static data.

24. The cryptographic key split combiner of claim 22, wherein the means for updating the static data includes means for modifying a prime number divisor of the static data.

25. The cryptographic key split combiner of claim 1, wherein said plurality of key split generators includes a biometric split generator for generating a biometric key split based on biometric data.

26. The cryptographic key split combiner of claim 25, wherein said biometric split generator includes means for generating a random sequence based on the biometric data.

27. The cryptographic key split combiner of claim 25, wherein said biometric split generator includes means for generating a pseudorandom sequence based on the biometric data.

28. The cryptographic key split combiner of claim 25, wherein said biometric split generator includes means for generating a key split based on biometric data vectors and on biometric combiner data.

29. The cryptographic key split combiner of claim 25, wherein said biometric split generator includes means for generating a key split based on the biometric data and on static data.

30. The cryptographic key split combiner of claim 29, further including means for updating the static data.

31. The cryptographic key split combiner of claim 30, wherein the means for updating the static data includes means for modifying a prime number divisor of the static data.

32. The cryptographic key split combiner of claim 1, wherein the cryptographic key is a stream of symbols.

33. The cryptographic key split combiner of claim 1, wherein the cryptographic key is at least one symbol block.

34. The cryptographic key split combiner of claim 1, wherein the cryptographic key is a key matrix.

35. A process for forming cryptographic keys, comprising:

a) generating a plurality of cryptographic key splits from seed data; and

b) randomizing the cryptographic key splits to produce a cryptographic key.

36. The process of claim 35, wherein generating a plurality of cryptographic key splits includes generating a random key split based on reference data.

37. The process of claim 36, wherein generating a random key split includes generating a random sequence based on the reference data.

38. The process of claim 36, wherein generating a random key split includes generating a pseudorandom sequence based on the reference data.

39. The process of claim 36, wherein generating a random key split includes generating a key split based on the reference data and on chronological data.

40. The process of claim 36, wherein generating a random key split includes generating a key split based on the reference data and on static data.

41. The process of claim 40, further including updating the static data.

42. The process of claim 41, wherein updating the static data includes modifying a prime number divisor of the static data.

43. The process of claim 35, wherein generating a plurality of cryptographic key splits includes generating a token key split based on label data.

44. The process of claim 43, further comprising reading the label data from a storage medium.

45. The process of claim 43, wherein the label data includes user authorization data.

46. The process of claim 43, wherein generating a token key split includes generating a random sequence based on the label data.

47. The process of claim 43, wherein generating a token key split includes generating a pseudorandom sequence based on the label data.

48. The process of claim 43, wherein generating a token key split includes generating a key split based on the label data and on organization data.

49. The process of claim 43, wherein generating a token key split includes generating a key split based on the label data and on static data.

50. The process of claim 49, further including updating the static data.

51. The process of claim 50, wherein updating the static data includes modifying a prime number divisor of the static data.

52. The process of claim 35, wherein generating a plurality of cryptographic key splits includes generating a console key split based on maintenance data.

53. The process of claim 52, wherein generating a console key split includes generating a random sequence based on the maintenance data.

54. The process of claim 52, wherein generating a console key split includes generating a pseudorandom sequence based on the maintenance data.

55. The process of claim 52, wherein generating a console key split includes generating a key split based on previous maintenance data and on current maintenance data.

56. The process of claim 52, wherein generating a console key split includes generating a key split based on the maintenance data and on static data.

57. The process of claim 56, further including updating the static data.

58. The process of claim 56, wherein the updating the static data includes modifying a prime number divisor of the static data.

59. The process of claim 35, wherein generating a plurality of cryptographic key splits includes generating a biometric key split based on biometric data.

60. The process of claim 59, wherein generating a biometric key split includes generating a random sequence based on the biometric data.

61. The process of claim 59, wherein generating a biometric key split includes generating a pseudorandom sequence based on the biometric data.

62. The process of claim 59, wherein generating a biometric key split includes generating a key split based on biometric data vectors and on biometric combiner data.

63. The process of claim 59, wherein generating a biometric key split includes generating a key split based on the biometric data and on static data.

64. The process of claim 63, further including updating the static data.

65. The process of claim 63, wherein updating the static data includes modifying a prime number divisor of the static data.

66. The process of claim 35, further including determining access conditions to a software program through selection of the generated key splits, wherein the software program includes component objects.

67. The process of claim 36, further including generating an asymmetrical key split based on one of the plurality of cryptographic key splits and on prime data.

68. A cryptographic key, formed by the process of claim 35.

69. The cryptographic key of claim 68, including a stream of symbols.

70. The cryptographic key of claim 68, including at least one symbol block.

71. The cryptographic key of claim 68, including a key matrix.

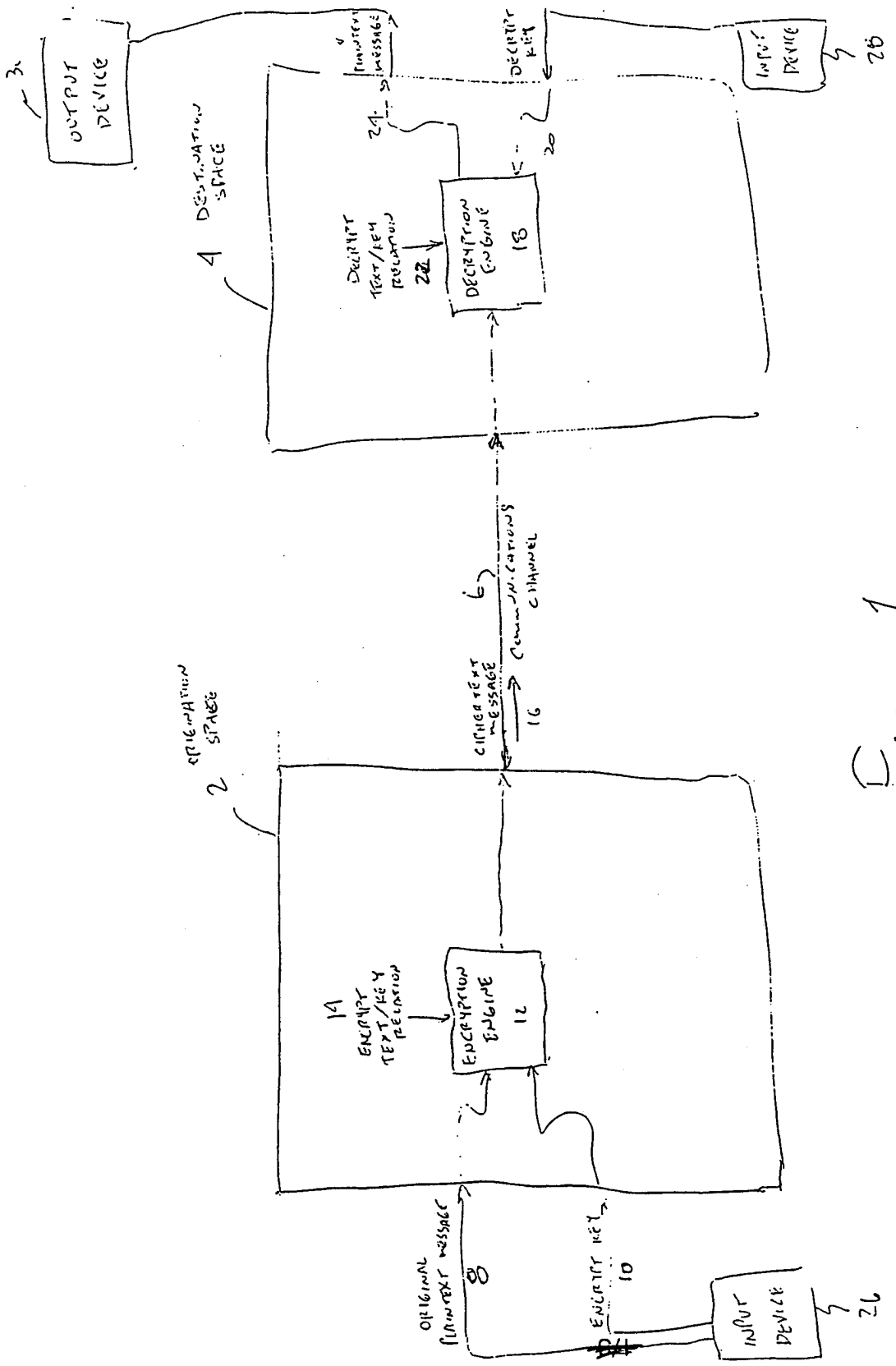

3
OUTPUT DEVICE
4
DECRYPTION SPACE
TRANSFERRED MESSAGE
DECRYPTION KEY
INPUT DEVICE
26
24
20
DECRYPT TEXT/KEY RELATIONSHIP
22
DECRYPTION ENGINE
18
6
CIPHERTEXT MESSAGE
16
COMMUNICATIONS CHANNEL
ENCRYPTION SPACE
2
14
ENCRYPT TEXT/KEY RELATIONSHIP
ENCRYPTION ENGINE
12
ORIGINAL PLAINTEXT MESSAGE
8
ENCRYPTION KEY
10
INPUT DEVICE
28

FIG. 1

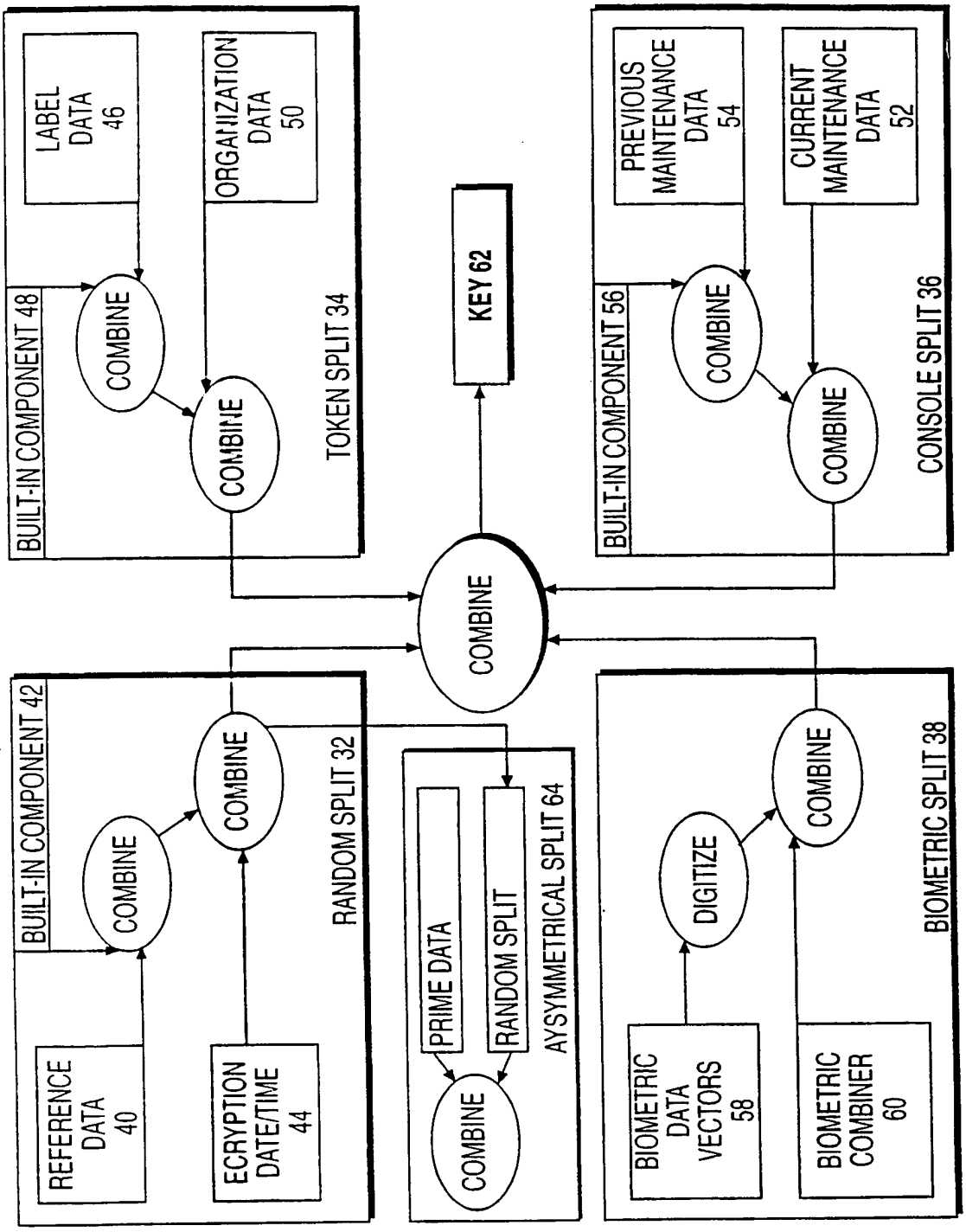

BUILT-IN COMPONENT 48
LABEL DATA 46
ORGANIZATION DATA 50
COMBINE
COMBINE
TOKEN SPLIT 34
KEY 62
BUILT-IN COMPONENT 56
PREVIOUS MAINTENANCE DATA 54
CURRENT MAINTENANCE DATA 52
COMBINE
COMBINE
CONSOLE SPLIT 36
COMBINE
BUILT-IN COMPONENT 42
REFERENCE DATA 40
COMBINE
ECRYPTION DATE/TIME 44
COMBINE
RANDOM SPLIT 32
PRIME DATA
RANDOM SPLIT
COMBINE
AYSYMMETRICAL SPLIT 64
BIOMETRIC DATA VECTORS 58
DIGITIZE
BIOMETRIC COMBINER 60
COMBINE
BIOMETRIC SPLIT 38

FIG. 2

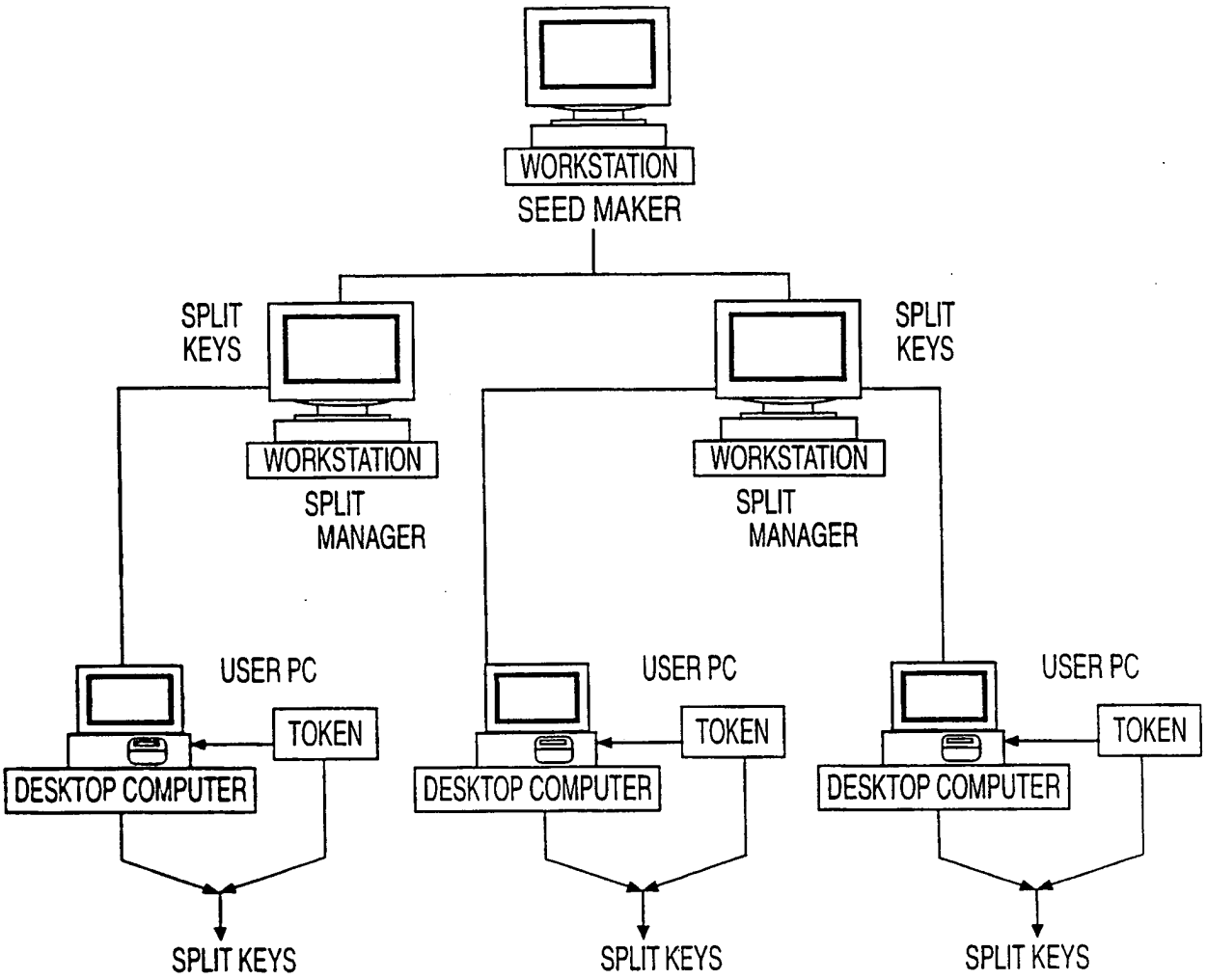

WORKSTATION
SEED MAKER
SPLIT KEYS
SPLIT KEYS
WORKSTATION
SPLIT MANAGER
WORKSTATION
SPLIT MANAGER
USER PC
USER PC
USER PC
TOKEN
TOKEN
TOKEN
DESKTOP COMPUTER
DESKTOP COMPUTER
DESKTOP COMPUTER
SPLIT KEYS
SPLIT KEYS
SPLIT KEYS

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/11304

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) :H04L 9/00
US CL : 380/42
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 380/42,9,21,28,43,44,45,46,47,49,50,59.

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,208,853 A (ARMBRUSTER et al) 04 May 1993, see abstract.	1-71
A	US 5,375,169 A (SCHEIDT et al) 20 December 1994, see abstract.	1-71
A	US 5,535,276 A (GANESAN) 09 July 1996, see abstract.	1-71
A	US 5,557,678 A (GANESAN) 17 September 1996, see abstract.	1-71

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
22 NOVEMBER 1997

Date of mailing of the international search report
13 FEB 1998

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
BERNARR EARL GREGORY
Telephone No. (703) 306-4153